МГУ имени Ломоносова

Практикум на ЭВМ, задание №1

Авторы: Елисеев 311 группа, Суворов 311 группа.

23.03.2019

Описание задания

Дана информация о:

1) закупках (поставки яблок и карандашей два раза в месяц);

supply.csv

Файл состоит из записей вида:

date	apple	pen
2006-01-01	2738	400
2006-01-02	2998	310
2006-01-03	3613	370

(число яблок и карандашей, поступивших на склад. Они поступают туда 1-го и 15-го числа каждого месяца)

2) Продажах (лог транзакций по записи на каждую проданную позицию)
Файл состоит из записей вида

data	sku_num
2006-01-01	MS-s4-ap-617233…
2006-01-01	MS-s4-ap-df379…
2006-01-02	MS-s4-pe-728hd…

- **date** - столбец обозначающий дату
- **sku_num** стобец содержащий идентификационный номер
- **MS -** название штата
- **s4 –** магазин
- **ap/pe -** наименование товара яблоко ручка
- **617233…** - идентификационный код

3) Инвентарь (месячные данные общего количества яблок и ручек на складе);

inventoru.csv

Файл состоит из записей вида

date	apple	pen
2006-01-31	5000	602
2006-02-28	6000	891
2006-03-31	8000	1030

data - столбец обозначающий дату
apple - стобец обозначающий количество яблок хранящихся на складе на конец месяца

pen - стобец обозначающий количество ручек хранящихся на складе на конец месяца

Учет товаров происходил раз в месяц
Нужно

1) Состояние склада на каждый день
2) Месячные данные о количестве сворованного товара
3) Агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году

Теория

Названия входных файлов можно условно поделить на две части: префикс, который имеет вид “<Штат>-<Название магазина>-” и суффикс **“supply.csv”, “cell.csv”, “inventory.csv”**. Мы отдельно работаем с каждым суффиксом, с помощью которого получаем доступ к три файлам: закупки, продажи и инвентарь. Сначала подсчитаем число яблок и ручек, проданных за каждый день. Для этого в файле **sell.csv** просто
подсчитываем число строк содержащих “ре”/”ар” для каждого дня. Далее составляем общую таблицу вида

date	apple	pen
2006-01-01	85	9
2006-01-02	97	10
2006-01-03	96	8

Состояние склада на каждый день

Мы знаем что поступления на склад происходят только раза в месяц Для дальнейшей работы с наборами данных мы расширим **inventory.csv** заполнив недостающие значения нулями. Отсюда вычтем то, что было продано, т.е. 1 и 15 числа будут положительные значения в таблице в остальные дни – отрицательные, т.к. ничего на склад не поступало, были только продажи. Далее просуммируем за каждый месяц - получим состояние склада на каждый день

Месячные данные о количестве сворованного товара

Чтобы получить значения сворованного товара за месяц достаточно вычесть из вычисленного реального
состояния склада в конце месяца количество товара указанное в инвентаре.

Агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году

Добавляем индексы штата и года в список **statistics**. Суммируем по ним все продажи и количество сворованного товара и конкатенируем в единый **DataFrame**.

Подход к решению

- **DataFrame.set_index(keys, inplace=False)**
  - Устанавливает индексирование в наборе данных
  - **keys** - столбец отвечающий за индексирование
  - **inplace** – **False** – создавать/ **True** - не создавать новый объект в результате применения операции - работать со старым
  - Библиотека **pandas**
- **DataFrame.groupby(by=None)**
  - Группирует данные по заданному аргументу
  - **by** - столбец относительно которого будет осуществляться перегруппировка
  - Библиотека **pandas**
- **DataFrame.agg(func)**
  - Агрегирует данные по заданному аргументу, т.е. имеющуюся в наборе данных информацию распределяет по новым столбцам по некоторому правилу
  - **func** - функция используемая для агрегирования данных
  - Библиотека **pandas**
- **DataFrame.reindex_like(other)**
  - Возвращает набор данных с индексированием подобным объекту аргументу
  - **other** - объект для копирования его индексирования
  - Библиотека **pandas**
- **DataFrame.fillna(value=None, method=None, axis=None)**
  - Заполняет ячеки некоторыми значениями
  - **value** - значение которым заполняются пустые ячейки
  - **method** - метод заполнения – (‘**ffill’** - ячейки заполняются значениями из вышестоящих ячеек)
  - Библиотека **pandas**
- **DataFrame.shift(periods=1)**
  - Сдвигает набор данных на указанное в значение
  - **periods** - на сколько сдвинуть
  - Библиотека **pandas**
- **DataFrame.join(other, lsuffix=’’,rsuffix=’’)**
  - Присоединяет к набору данных столбцы, добавляет соответствующие суффиксы к названиям столбцов
  - **other** - другой набор данных при помощи которого осуществляется присоединение
  - **lsuffix** - суффикс для левых столбцов
  - **rsuffix** - суффикс для правых столбцов
  - Библиотека **pandas**
- **Series.map(arg)**
  - При помощи **arg** редактирует соответствующий столбец **arg** - аргумент
  - Библиотека **pandas**
- **DataFrame.to_csv(path=None)**
  - Создает на основе входного набора данных файл с именем **path**

    - **path** - имя файла
    - Библиотека **pandas**
- **os.path.join(path, *paths)**
    - Конкатенация и элементов
    - **path** - список элементов
    - Библиотека **os.path**
- **os.path.isfile(path)**
    - Возвращает **Thue,** если файл по указанному пути действительно существует
    - **path** - путь
    - Библиотека **os.path**
- **os.listdir(path)**
    - Возвращает список файлов находящихся в указанной директории
    - **path** - путь
    - Библиотека **os**
- **os.path.exists(path)**
    - Возвращает **True**, если указанный путь действительно существует
    - **path** - путь
    - Библиотека **os.path**
- **os.makedirs(path)**
    - Создает директорию с заданным именем. Если такая директория уже существует, выбрасывается исключение
    - **path** - имя создаваемой директории
    - Библиотека **os**
- **shutil.rmtree(path)**
    - Рекурсивно удаляет содержимое указанной директории Если её нет выбрасывается исключение
    - **path** путь к подлежащей удалению директории
    - Библиотека **shutil**

Инструкции по запуску

1. Поместить в папку **‘input’** файлы для обработки
2. **Cell -> Run All**
3. Результат будет находиться в папке **’output’**

Необходимое ПО

1. Jupiter Notebook

Библиотеки:

1. **Pandas** - Для работы с временными рядами
2. **Numpy** - Для работы с массивами
3. **OS** - Для работы с файлами и директориями
4. **Shutil** - Для удаления директории и её содержимого

Участники:

Суворов 311 группа, Елисеев 311 группа – все этапы работы были выполнены совместно